

Data Analytics, Autumn 2024

Assignment 3

DEADLINE - 15.11.24 – 11:59 PM (Hard Deadline)

The task of this assignment is to build a recommendation system that suggests top actors to users based on their movie preferences. You will work with the MovieLens dataset, extending it by collecting information on the top actors for each movie. Using this enriched data, you will generate user-actor ratings and apply a recommendation algorithm to recommend actors to users. The goal is to identify actors that align with each user's tastes by leveraging collaborative or content-based filtering techniques and evaluating the system with ranking-based metrics like Precision@k.

The [dataset](#) contains movie ratings from multiple users, with each row representing a user's rating for a specific movie.

Task 1: Finding Top Actors for Each Movie

- For each movie title in the MovieLens dataset, find the **top actors** in that movie.
- You can use a web scraping tool (e.g., BeautifulSoup, Scrapy) to retrieve the actor information.
- Document the methods used for data scraping, including any API requests or HTML parsing techniques.
- Store the movie-actor relationships as a separate dataset for future use.

2. Task 2: Constructing the User-Actor Rating Matrix

- Using the user-movie ratings in the MovieLens dataset and the movie-actor associations, construct a **user-actor rating dataset**.
- Define each user's rating for an actor as the **average rating** they gave to the movies featuring that actor.
- Make sure to handle cases where a user has rated multiple movies with the same actor by taking an average.
- After constructing the dataset, normalize or preprocess it if necessary (e.g., remove low-activity users or actors with few ratings).

3. Task 3: Choosing a Recommendation Algorithm

- Choose a suitable recommendation algorithm for recommending actors to users based on their preferences. Justify your choice of recommendation algorithm with respect to this specific dataset and task.

4. Task 4: Implementing the Recommendation System

- Implement the chosen recommendation algorithm to generate actor recommendations for a given user.
- Given a user ID, the system should output a ranked list of **top actors** that the user is likely to enjoy based on their previous ratings.

5. Task 5: Evaluation and Analysis

- Evaluate the performance of your recommendation system using appropriate metrics such as:
 - **Precision@k**: Measures the proportion of relevant actors in the top-k recommendations..

- **Recall@k**: Indicates the proportion of all relevant actors retrieved within the top-k recommendations..
 - **NDCG@k** (Normalized Discounted Cumulative Gain): Assesses the quality of ranking by considering the position of relevant actors in the recommended list.
-

Submission Requirements

Before evaluating the performance, we will first conduct a code **plagiarism** check. For evaluation, we will use different metrics to judge the efficiency of the algorithm. Additionally, we will measure the running time of each implementation.

For this assignment, each team must submit a **ZIP** file having a Python notebook (.ipynb) and a report **pdf** file. The notebook should include clearly labeled headings and proper comments for each cell for the execution. The report must have all the metrics, plots, and other important output values. Highlight any **advanced modifications** that you've added to the algorithm for better performance and why it would be beneficial for the task.

We'll use MS Teams to accept the submissions. Only **one member** from each team should submit the assignment deliverables. The report should have the **names** and **roll numbers** of all the team members.

Report:

- **Data Collection and Preprocessing**: Describe the scraping process, any challenges faced, and the resulting movie-actor dataset.
- **User-Actor Rating Construction**: Explain how you calculated user-actor ratings from user-movie ratings.
- **Algorithm Selection**: Justify the recommendation algorithm you chose and describe how it was implemented.
- **Evaluation and Analysis**: Include metrics and a brief analysis of the model's performance, along with a discussion of strengths and limitations.
- (optional) If possible, include a visualization that showcases some aspect of your recommendation system (e.g., top-rated actors for a sample user, distribution of actor ratings, etc.).